In honor of Professor Paul Tseng, who went missing while on a kayak trip on the Jinsha river, China, on August 13, 2009, for his contributions to the theory and algorithms for large-scale optimization.

Abstract. In this paper, we introduce a flexible optimization framework for nuclear norm minimization of matrices with linear structure, including Hankel, Toeplitz and moment structures, and catalog applications from diverse fields under this framework. We discuss various first-order methods for solving the resulting optimization problem, including alternating direction methods, proximal point algorithm and gradient projection methods. We perform computational experiments to compare these methods on system identification problem and system realization problem. For the system identification problem, the gradient projection method (accelerated by Nesterov’s extrapolation techniques) and the proximal point algorithm usually outperform other first-order methods in terms of CPU time on both real and simulated data, for small and large regularization parameters respectively; while for the system realization problem, the alternating direction method, as applied to a certain primal reformulation, usually outperforms other first-order methods in terms of CPU time.

Key words. Rank minimization, nuclear norm, Hankel matrix, first-order method, system identification, system realization

1. Introduction. The matrix rank minimization problem, or minimizing the rank of a matrix subject to convex constraints, has recently attracted much renewed interest. This problem arises in many engineering and statistical modeling applications, where notions of order, dimensionality, or complexity of a model can be expressed by the rank of an appropriate matrix. Thus, choosing the “simplest” model that is consistent with observations or data often translates into finding a matrix with the smallest rank subject to convex constraints. Rank minimization is NP-hard in general, and a popular convex heuristic for it minimizes the nuclear norm of the matrix (the sum of the singular values) instead of its rank [17]. The regularized version of this problem can be written as

\[
\min_X \frac{1}{2} \| \mathbf{A}(X) - b \|_2^2 + \mu \| X \|_*,
\]

where \( X \in \mathbb{R}^{m \times n} \) is the optimization variable and \( \mathbf{A} : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}^p \) is a linear map, \( b \in \mathbb{R}^p \), and \( \mu > 0 \) is the tradeoff parameter between the nuclear norm and the least squares fitting error. Problem (1.1) has been widely studied and recently a variety of efficient algorithms have been developed [5, 9, 26, 29, 30, 37, 51]. A special case of this problem is the matrix completion problem [7, 8] which has applications...
in collaborative filtering and machine learning. In this problem the measurements are simply a subset of the entries of the matrix. The majority of existing work on algorithms for problem (1.1) has concentrated on this special case.

In this paper, we focus on problems where we need to find a matrix \( X \) that, in addition to being low-rank, is required to have a certain linear structure, for example, (block-)Hankel, (block-)Toeplitz, or moment structure. Hankel (and Toeplitz) structures arise in dynamical systems problems discussed in Section 1.1, while moment structure comes up in Lasserre relaxations for minimizing polynomials [27]. We consider problem (1.1), and represent the desired structure by a linear map \( X = \mathcal{H}(y) \), where \( y \) is our optimization variable. Note that if \( \mathcal{H}(y) \) is a moment matrix we need to add the constraint \( \mathcal{H}(y) \succeq 0 \).

1.1. Motivating applications.

1.1.1. Applications in linear dynamical systems. Linear time-invariant (LTI) systems have a long and successful history in modeling dynamical phenomena in many fields, from engineering to finance. The goal of fitting an LTI model to observed data gives rise to different classes of optimization problems, depending on whether the model is parametric or black-box, given in time or frequency domain, deterministic or stochastic, as well as on the type of data, e.g., input-output or state measurements (see, e.g., [12,19,33]). In all these cases, picking the appropriate model order or complexity, and understanding its tradeoff with the fitting or validation errors is crucial. In the problems described in this section, the system order or complexity can be expressed as the rank of a Hankel-type matrix. We discuss some of these problems in more detail in Sections 4 and 5.

**Minimal system realization with time-domain constraints.** Consider the problem of designing a discrete-time, linear time-invariant (LTI) dynamical system, directly from convex specifications on the system’s response in the time domain; see, e.g., [18, 31]. Such a problem arises in designing filters for signal processing and control applications. The objective is to trade-off the order of the linear system with how well the specifications are met. A low-order design is desired since in practice, it translates into a system that is easier and cheaper to build and analyze. Typical specifications are desired rise-time, settling-time, slew-rate, and overshoot of the filter’s response to a step input signal. These specifications can be expressed as upper and lower bounds on the step response over a fixed time horizon, say \( n \) time samples. Equivalently, they can be written in terms of the impulse response, which translate into linear inequality constraints on the entries of a Hankel matrix whose rank corresponds to the system order or McMillan degree; see, e.g., [50]. Using the nuclear norm heuristic for rank, we get

\[
\min \| \mathcal{H}(y) \|_*, \quad \text{s.t.} \quad l_i \leq \sum_{k=1}^i y_k \leq b_i, \quad i = 1, \ldots, n, \tag{1.2}
\]

where the optimization variable is \( y \in \mathbb{R}^{2n} \) with \( y_i \) corresponding to the value of the impulse response at time \( i \), \( l_i \) and \( b_i \) denoting the bounds on the step response given by the specifications, and \( \mathcal{H}(y) \) denoting an \( n \times n \) Hankel matrix; see [18] for more details. Notice that this problem is not exactly in the form of (1.1); we shall discuss how algorithms proposed in this paper can be extended to tackle this model in Appendix A.

**Minimal partial realization.** A related problem in linear system theory is the minimal partial realization problem for multi-input, multi-output systems: given a
sequence of matrices $H_k$, $k = 1, \ldots, n$, find a minimal state space model, described by a 3-tuple of appropriately sized matrices $(A, B, C)$ such that $H_k = CA^{k-1}B$ [50, Chapter 6]. This problem can be viewed as a more general version of the above realization problem which handled a single input and a single output. In this problem, the order of the system (minimal size of a state-space representation) is equal to the rank of a block-Hankel matrix consisting of the $H_k$; see [32, section II.A].

**Input-output system identification (system ID).** Identifying a linear dynamical system given noisy and/or partial observations of its inputs and outputs, also related to time-series analysis, is a fundamental problem studied in a variety of fields [54,55], including signal processing, control and robotics; see, e.g., [11,33]. We will discuss this problem and a Hankel rank formulation for it in detail in Section 4.

An interesting variation of this problem is a case where the output information is limited to a few time points, for example the case with a switched output briefly mentioned in [46, Section 1]. In this setup, our observations are the system output sampled at a fixed time $T$, after an input signal is applied from $t = 0$ to $t = T$. We make output measurements for several different input signals, observing $y_i(T) = \sum_{t=0}^{T} a_i(T - j) h(t)$, where the vector $a_i$ is the $i$th input signal and $h(t)$ denotes the impulse response. Writing this compactly as $y = Ah$ where $A_{ij} = a_i(T - j)$ and $h = [h(0) \ldots h(T)]^T$ is the optimization variable, and replacing rank of the Hankel matrix with its nuclear norm, we get a problem of the form (1.1). However, unlike the main system ID problem, in this setup the linear map $A$ has a nullspace so the fitting error term is not strongly convex; as discussed later, this affects the choice of algorithm for solving the problem computationally.

**Stochastic realization.** Another fundamental problem in linear system theory is finding a minimal stochastic ARMA (autoregressive moving average) model for a vector random process, given noisy and/or partial estimates of process covariances [12,34]. The minimal order is the rank of a block-Hankel matrix consisting of the exact covariances. This problem is discussed in detail in Section 5.

1.1.2. Other applications.

**Shape from moments estimation.** Consider a polygonal region $P$ in the complex plane with ordered vertices $z_1, \ldots, z_m$. Complex moments of $P$ are defined as $\tau_k := k(k-1) \int_P z^{k-2} dx dy$, $\tau_0 = \tau_1 = 0$, and can be expressed as $\tau_k = \sum_{i=1}^m a_i z_i^k$, where $m$ is the number of vertices and $a_i$ are complex constants. The problem of determining $P$ given its complex moments has been studied in [16,38,49], and arises in many applications such as computer tomography, where X-ray is used to estimate moments of mass distribution, and geophysical inversion, where the goal is to estimate the shape of a region from external gravitational measurements. Note that the number of vertices is equal to the rank of the Hankel matrix consisting of the moments [16,23]. In practice, often only noisy or partial measurements of the complex moments are available, and the challenge is to find a polygon with the minimum number of vertices that is consistent with the measurements. Formulating this problem as a rank minimization problem, we propose using the nuclear norm heuristic for rank. This leads to a problem of the form (1.1), where the optimization variable is the vector of complex moments $\tau$.

**Moment matrix rank minimization for polynomial optimization.** Suppose $p(x)$, $x \in \mathbb{R}^n$ is a polynomial of degree $d$. Denote the corresponding moment matrix by $M(y)$, where $y$ is the vectors of moments, i.e., $y_i$ corresponds to the $i$th monomial; see [27,28]. Moment matrices are important in Lasserre’s hierarchy of relaxations for polynomial optimization, where a condition on the rank of the moment matrix
in successive relaxations in the hierarchy determines whether the relaxation is exact; see, e.g., [28, Section 5].

In the dual problem of representing a polynomial as a sum of squares of other polynomials [44], the rank of the coefficients matrix equals the minimum number of squared polynomials in the representation, thus the nuclear norm (or trace) penalty helps find simpler representations. Note that in these problems, we also have an additional positive semidefinite constraint on the desired matrix.

Further applications. Another application arises in video inpainting in computer vision, where features extracted from video frames are interpolated by finding a low-rank completion to a Hankel matrix, and help reconstruct the missing frames or occluded parts of a frame [13].

Finally, our problem formulation also gives a relaxation for the problem known as the Structured Total Least Squares problem [10] studied extensively in the controls community; see [35,36]. Our algorithms are thus applicable to this set of problems as well. A variety of applications are discussed in [36].

1.2. Our contributions. We introduce a flexible optimization framework for nuclear norm minimization of linearly structured matrices, including Hankel, Toeplitz, and moment matrices. We identify and catalog applications from diverse fields, some of which have not been studied from this perspective before; for example, the shape from moments estimation problem. In view of the wide applicability of the model (1.1), it is important to find efficient algorithms for solving it. Along this direction, recently, Liu and Vandenberghe [31] proposed an interior-point method for solving a reformulation of problem (1.1), where they used the SDP representation for the nuclear norm and exploited the problem structure to efficiently solve the Newton system. They applied their algorithm to the system identification and system realization problems mentioned above. The cost per iteration of the algorithm grows roughly as $O(pqn^2)$, where $y \in \mathbb{R}^n$ and $X = H(y) \in \mathbb{R}^{p \times q}$, $q \leq p \leq n$.

In this paper, we derive various primal and dual reformulations of problem (1.1) (with $X = H(y)$), and propose several first-order methods for solving the reformulations. In particular, we show that the alternating direction method and the proximal point algorithm can be suitably applied to solve reformulations of (1.1). These methods have been widely used in the literature recently for solving (1.1), when no linear structure is imposed on $X$; see, e.g., [30, 56]. We discuss implementation detail of these methods in Section 3.1 and Section 3.2. For these methods, typically, each iteration involves a singular value decomposition whose cost grows as $O(p^2q)$ where $X = H(y) \in \mathbb{R}^{p \times q}$ and $p < q$; see, e.g., [24, Page 254]. Next, in Section 3.3, assuming that $A^*A$ is invertible, we show that the (accelerated) gradient projection algorithms can be efficiently applied to solve a dual reformulation of (1.1). In this approach, each iteration also involves a singular value decomposition and there is an explicit upper bound on the number of iterations required to attain a certain accuracy. This solution approach has been considered recently in [39] for solving the system identification problem.

To demonstrate the computational efficiency of our algorithms, we apply them to solve the input-output system identification problem and the stochastic system realization problem. For the system identification problem, we consider both simulated data and real data from the DaISy database [11]. Our computational results show that the accelerated gradient projection algorithm and the proximal point algorithm usually outperform other first-order methods for this application in terms of CPU time for small and large regularization parameter, respectively. We also observe that
these methods significantly outperform the interior point implementation proposed in [31]. For the system realization problem, we consider only simulated data, and our computational results show that the alternating direction method, as applied to a certain primal reformulation of (1.1), usually outperforms other first-order methods in terms of CPU time.

The rest of this paper is organized as follows. In Section 1.3, we introduce notations used in this paper. We then derive primal and dual reformulations of (1.1) in Section 2 and discuss several first-order methods for solving the reformulations in Section 3. In Section 4 and Section 5, we present computational results of our first-order methods for solving the system identification problem and system realization problem, respectively. We give some concluding remarks in Section 6. Finally, we discuss an alternative formulation for modeling the structured matrix rank minimization problem in Appendix A, and provide a convergence proof that covers all versions of proximal alternating direction methods used in this paper in Appendix B.

1.3. Notation. In this paper, $\mathbb{R}^n$ denotes the $n$-dimensional Euclidean space. For a vector $x \in \mathbb{R}^n$, $\|x\|$ denotes the Euclidean norm of $x$. The set of all $m \times n$ matrices with real entries is denoted by $\mathbb{R}^{m \times n}$. For any $A \in \mathbb{R}^{m \times n}$, $\|A\|$ denotes the spectral norm of $A$, $\|A\|_F$ denotes the Frobenius norm of $A$, $\|A\|_*$ denotes the nuclear norm of $A$, which is the sum of all singular values of $A$, and $\text{vec}(A)$ denotes the column vector formed by stacking columns of $A$ one by one. For two matrices $A$ and $B$ in $\mathbb{R}^{m \times n}$, $A \circ B$ denotes the Hadamard (entry-wise) product of $A$ and $B$. If a symmetric matrix $A$ is positive semidefinite, we write $A \succeq 0$. Linear maps will be denoted by scripted letters. For a linear map $\mathcal{A} : \mathbb{R}^{m \times n} \to \mathbb{R}^p$, $\mathcal{A}^*$ denotes the adjoint of $\mathcal{A}$, $\|\mathcal{A}\|$ denotes the spectral norm of $\mathcal{A}$, while $\sigma_{\max}(\mathcal{A})$ and $\sigma_{\min}(\mathcal{A})$ denote the maximum and minimum singular value of $\mathcal{A}$, respectively. Finally, we denote the identity matrix and identity map by $I$ and $\mathcal{I}$ respectively, whose dimensions should be clear from the context.

2. Basic problem formulations. Consider the following general Hankel matrix nuclear norm minimization problem:

$$ v := \min_y f(y) := \frac{1}{2} \|\mathcal{A}(y) - b\|^2 + \mu \|\mathcal{H}(y)\|_*, $$

(2.1)

where $\mathcal{A} : \mathbb{R}^{m \times n(j+k-1)} \to \mathbb{R}^p$ is a linear map, $b \in \mathbb{R}^p$, $y = (y_0 \cdots y_{j+k-2})$ is an $m \times n(j + k - 1)$ matrix with each $y_i$ being an $m \times n$ matrix for $i = 1, \ldots, j + k - 2$, and $\mathcal{H}(y) := H_{m,n,j,k}(y)\mathcal{Y}$ with

$$ H_{m,n,j,k}(y) := \begin{pmatrix} y_0 & y_1 & \cdots & y_{k-1} \\ y_1 & y_2 & \cdots & y_k \\ \vdots & \vdots & \ddots & \vdots \\ y_{j-1} & y_j & \cdots & y_{j+k-2} \end{pmatrix} \in \mathbb{R}^{mj \times nk}, $$

and $\mathcal{Y} \in \mathbb{R}^{nk \times q}$. We assume without loss of generality that $\sigma_{\max}(\mathcal{Y}) \leq 1$. In this section, we derive primal and dual reformulations of (2.1)

First, using the substitutions $Y = -\mathcal{H}(y)$ and $z = b - \mathcal{A}(y)$, problem (2.1) can be reformulated as

$$ \min_{Y,z,Y} p(Y, z) := \frac{1}{2} \|z\|^2 + \mu \|Y\|_*, $$

(2.2)

s.t. $Y + \mathcal{H}(y) = 0,$

$z + \mathcal{A}(y) = b.$
From the reformulation (2.2), we can easily write down the Lagrange dual to (2.2) (and hence, equivalently, to (2.1)) as follows.

\[
v = \min_{Y,z,y} \max_{\gamma,A} \left\{ \frac{1}{2} \|z\|^2 + \mu \|Y\|_* - \langle \Lambda, Y + \mathcal{H}(y) \rangle - \langle \gamma, z + \mathcal{A}(y) - b \rangle \right\}
\]

\[
= \max_{\gamma,A} \min_{Y,z,y} \left\{ \frac{1}{2} \|z\|^2 + \mu \|Y\|_* - \langle \Lambda, Y + \mathcal{H}(y) \rangle - \langle \gamma, z + \mathcal{A}(y) - b \rangle \right\}
\]

\[
= \max_{\gamma,A} \min_{Y,z,y} \left\{ \frac{1}{2} \|z\|^2 - \langle \gamma, z \rangle + \mu \|Y\|_* - \langle \Lambda, Y \rangle - \langle \mathcal{H}^*(\Lambda) + \mathcal{A}^*(\gamma), y \rangle + \langle b, \gamma \rangle \right\}
\]

\[
= \max_{\gamma,A} \left\{ \frac{1}{2} \|\gamma\|^2 + \langle \gamma, b \rangle : \mathcal{H}^*(\Lambda) + \mathcal{A}^*(\gamma) = 0, \Lambda^T \Lambda \preceq \mu^2 I \right\},
\]

where the second equality holds because of the compactness of the spectral norm ball [47, Corollary 28.2.2]. The dual problem can thus be rewritten as the following minimization problem.

\[
\min_{\gamma,A} \quad d(\gamma) := \frac{1}{2} \|\gamma\|^2 - b^T \gamma \\
\text{s.t.} \quad \mathcal{H}^*(\Lambda) + \mathcal{A}^*(\gamma) = 0, \quad \Lambda^T \Lambda \preceq \mu^2 I. \tag{2.3}
\]

Alternatively, noting the fact that the nuclear norm is just the dual norm of the spectral norm, one can derive a reduced dual problem as follows:

\[
v = \min_y \frac{1}{2} \|\mathcal{A}(y) - b\|^2 + \mu \|\mathcal{H}(y)\|_* = \min_y \max_{\Lambda^T \Lambda \preceq \mu^2 I} \frac{1}{2} \|\mathcal{A}(y) - b\|^2 - \langle \Lambda, \mathcal{H}(y) \rangle
\]

\[
= - \min_{\Lambda^T \Lambda \preceq \mu^2 I} d_2(\Lambda) := \sup_y \left\{ \langle \Lambda, \mathcal{H}(y) \rangle - \frac{1}{2} \|\mathcal{A}(y) - b\|^2 \right\}, \tag{2.4}
\]

where the third equality holds because of the compactness of the spectral norm ball [47, Corollary 37.3.2]. In the special case when \(\mathcal{A}^*\mathcal{A}\) is invertible, the function \(d_2\) has a closed form representation. Indeed, in this case, writing \(\mathcal{R}(\Lambda) := (\mathcal{A}^*\mathcal{A})^{-1}\mathcal{H}^*(\Lambda)\) and \(\bar{b} := (\mathcal{A}^*\mathcal{A})^{-1}\mathcal{A}^*b\), we obtain that

\[
d_2(\Lambda) = \frac{1}{2} \left( (\mathcal{H}^*(\Lambda), \mathcal{R}(\Lambda)) + 2\langle \mathcal{H}^*(\Lambda), \bar{b} \rangle + \langle \mathcal{A}^*b, \bar{b} \rangle - \|b\|^2 \right).
\]

Hence, when \(\mathcal{A}^*\mathcal{A}\) is invertible, the reduced dual problem (2.4) is equivalent to

\[
\min_{\Lambda} \frac{1}{2} \langle \mathcal{H}^*(\Lambda), \mathcal{R}(\Lambda) \rangle + \langle \mathcal{H}^*(\Lambda), \bar{b} \rangle + \frac{1}{2} \langle \mathcal{A}^*b, \bar{b} \rangle - \frac{1}{2} \|b\|^2 \tag{2.5}
\]

Before ending this section, we derive an upper bound on the spectral norm of \(\mathcal{H}^*\). Notice that \(\mathcal{H}^*(\Lambda) = H_{m,n,j,k}^*(\Lambda^T^T)\), where for any \(W \in \mathbb{R}^{m \times nk}\)

\[
H_{m,n,j,k}^*(W) = H_{m,n,j,k}^* \begin{pmatrix} w_{00} & w_{01} & \cdots & w_{0,k-1} \\ w_{10} & w_{11} & \cdots & w_{1,k-1} \\ \vdots & \vdots & \ddots & \vdots \\ w_{j-1,0} & w_{j-1,1} & \cdots & w_{j-1,k-1} \end{pmatrix} = \begin{pmatrix} w_{00} & w_{01} + w_{10} & w_{02} + w_{11} + w_{20} & \cdots & w_{j-1,0} + w_{j-1,1} + \cdots + w_{j-1,k-1} \end{pmatrix} \in \mathbb{R}^{m \times n(j+k-2)}. \tag{2.6}
\]
It follows from (2.6) that
\[
\|H_{m,n,k}(W)\|_F^2 = \|w_00\|_F^2 + \|w_{01} + w_{10}\|_F^2 + \|w_{02} + w_{11} + w_{20}\|_F^2 + \cdots + \|w_{j-1,k-1}\|_F^2 \\
\leq \|w_00\|_F^2 + 2\|w_{01}\|_F^2 + \|w_{10}\|_F^2 + \cdots + \|w_{j-1,k-1}\|_F^2 \leq r\|W\|_F^2,
\]
where \(r := \min\{j, k\}\). Combining this estimate with \(\sigma_{\text{max}}(Y) \leq 1\), we obtain that
\[
\|H^*(\Lambda)\|_F^2 \leq r\|AY^T\|_F^2 \leq r\|A\|_F^2,
\]
and thus the spectral norm of \(H^*\) is less than or equal to \(\sqrt{r}\).

3. Algorithms. In this section, we discuss several first-order methods for solving (2.2) and (2.3) (and hence (2.1)).

3.1. Alternating direction methods. In this section, we discuss how the alternating direction method (ADM) can be applied to solve (2.2) and (2.3). To apply the ADM for solving (2.2), we first introduce the augmented Lagrangian function
\[
L_\beta(Y, z, y, \gamma, \Lambda) = \frac{1}{2}\|z\|_2^2 + \mu\|Y\|_* - \langle A, Y + H(y) \rangle - \langle \gamma, z + A(y) - b \rangle + \frac{\beta}{2}\|Y + H(y)\|_F^2 + \frac{\beta}{2}\|z + A(y) - b\|^2
\]
for each \(\beta > 0\). In the classical ADM (see, e.g., [2, Section 3.4.4]), in each iteration, we minimize \(L_\beta\) with respect to \((Y, z)\) and then with respect to \(y\), followed by an update of the multiplier \((\gamma, \Lambda)\). While minimizing \(L_\beta\) with respect to \((Y, z)\) admits an easy closed form solution, minimizing \(L_\beta\) with respect to \(y\) does not usually have a simple closed form solution due to the complicated quadratic terms. One way to resolve this is to add a proximal term with norm induced by a suitable positive (semi-)definite matrix to “cancel” out the complicated parts. In this approach, we update
\[
y^{k+1} = \arg\min_y \left\{ L_\beta(Y^{k+1}, z^{k+1}, y, \gamma^k, \Lambda^k) + \frac{\beta}{2}\|y - y^k\|_{Q_0}^2 \right\},
\]
where \(\|\cdot\|_{Q_0}\) is the norm induced from the inner product \(x^TQ_0x\),
\[
Q_0 := \frac{1}{\sigma}I - (H^*H + A^*A) > 0 \text{ and } \sigma < \frac{1}{r + (\sigma_{\text{max}}(A))^2}
\]
so that \(\sigma\|H^*H + A^*A\| \leq \sigma(\|H^*H\| + \|A^*A\|) < 1\). The convergence analysis of this approach has been considered in [25], for example, in the context of variational inequalities \(^1\). For the sake of completeness, we discuss convergence of this approach in detail in the appendix. We now present our algorithm as follows.

**Primal ADM.**

**Step 0.** Input \((y^0, \gamma^0, \Lambda^0), \beta > 0\) and \(0 < \sigma < \frac{1}{r + (\sigma_{\text{max}}(A))^2}\).

**Step 1.** Compute the SVD
\[
-H(y^k) + \frac{\Lambda^k}{\beta} = U\Sigma V^T,
\]
where \(U\) and \(V\) have orthogonal columns, \(\Sigma\) is diagonal. Set
\[
Y^{k+1} = U \max\left\{ \Sigma - \frac{\mu}{\beta}I, 0 \right\} V^T, \quad z^{k+1} = \frac{1}{1 + \beta} (\gamma^k - \beta A(y^k) + \beta b),
\]
\[
y^{k+1} = y^k - \sigma \left( -\frac{1}{\beta} (H^*A + A^*\gamma^k) + H^*(H(y^k) + Y^{k+1}) + A^*(A(y^k) + z^{k+1} - b) \right),
\]
\[
\gamma^{k+1} = \gamma^k - \beta(z^{k+1} + A(y^{k+1}) - b), \quad \Lambda^{k+1} = \Lambda^k - \beta(Y^{k+1} + H(y^{k+1})).
\]

\(^1\)The main motivation of introducing the proximal terms in [25] is to weaken the imposed convergence conditions rather than for the sake of cancelation, as was more recently explained in [57].
Step 2. If a termination criterion is not met, go to Step 1.

The ADM can also be applied to solve the dual problem (2.3). In this approach, we make use of the following augmented Lagrangian function

$$l_\beta(\gamma, \Lambda, y) = \frac{1}{2} \|\gamma\|^2 - b^T \gamma + \langle y, \mathcal{H}^*(\Lambda) + \mathcal{A}^*(\gamma) \rangle + \frac{\beta}{2} \|\mathcal{H}^*(\Lambda) + \mathcal{A}^*(\gamma)\|^2,$$

for each $\beta > 0$. Notice that the minimizer of $l_\beta$ with respect to $\gamma$ is usually not easy to find and the minimizer with respect to $\Lambda$ does not always admit a simple closed form solution. Thus, as before, we add suitable proximal terms to cancel out complicated terms. More precisely, we update

$$\gamma^{k+1} := \arg\min_\gamma \left\{ l_\beta(\gamma, \Lambda^k, y^k) + \frac{\beta}{2} \|\gamma - \gamma^k\|^2_{Q_1} \right\},$$

$$\Lambda^{k+1} := \arg\min_{\Lambda^T \Lambda \preceq \mu^2 I} \left\{ l_\beta(\gamma^{k+1}, \Lambda, y^k) + \frac{\beta}{2} \|\Lambda - \Lambda^k\|^2_{Q_2} \right\},$$

where

$$Q_1 := \frac{1}{\sigma_1} I - \mathcal{A}^* \mathcal{A} > 0 \quad \text{and} \quad \sigma_1 < \frac{1}{\sigma_{\max}(\mathcal{A})^2}, \quad Q_2 := \frac{1}{\sigma_2} I - \mathcal{H} \mathcal{H}^* > 0 \quad \text{and} \quad \sigma_2 < \frac{1}{\mathcal{F}}.$$

so that $\sigma_1 \|\mathcal{A}^*\| < 1$ and $\sigma_2 \|\mathcal{H} \mathcal{H}^*\| < 1$. The algorithm is described as follows.

**Dual ADM.**

**Step 0.** Input $(y^0, \gamma^0, \Lambda^0)$, $\beta > 0$, $0 < \sigma_1 < \frac{1}{(\sigma_{\max}(\mathcal{A}))^2}$ and $0 < \sigma_2 < \frac{1}{\mathcal{F}}$.

**Step 1.** Set

$$\gamma^{k+1} = \frac{\sigma_1}{\sigma_1 + \beta} \left( b + \beta \gamma^k \right) - \beta \left( \frac{\mathcal{A}(y^k)}{\beta} + \mathcal{A}(\mathcal{H}^*(\Lambda^k) + \mathcal{A}^*(\gamma^k)) \right).$$

Compute the SVD

$$\Lambda^k = \sigma_2 \left( \frac{1}{\mathcal{F}} \mathcal{H}(y^k) + \mathcal{H}(\mathcal{H}^*(\Lambda^k) + \mathcal{A}^*(\gamma^k)) \right) = UV^T,$$

where $U$ and $V$ have orthogonal columns, $\Sigma$ is diagonal. Set

$$\Lambda^{k+1} = U \min \{\Sigma, \mu I\} V^T,$$

$$y^{k+1} = y^k + \beta(\mathcal{H}^*(\Lambda^{k+1}) + \mathcal{A}^*(\gamma^{k+1})).$$

**Step 2.** If a termination criterion is not met, go to Step 1.

From Theorem B.1 in the appendix, for the sequence $\{(Y^k, z^k, y^k, \gamma^k, \Lambda^k)\}$ generated from Primal ADM, $\{(Y^k, z^k, y^k)\}$ converges to a solution of the problem (2.2) while $\{(\gamma^k, \Lambda^k)\}$ converges to a solution of the problem (2.3). Similarly, for the sequence $\{(y^k, \gamma^k, \Lambda^k)\}$ generated from Dual ADM, $\{y^k\}$ converges to a solution of (2.1) and $\{\gamma^k, \Lambda^k\}$ converges to a solution of (2.3).

**3.2. Dual proximal point algorithm.** In this section, we discuss how the proximal point algorithm (PPA) can be applied to solve the dual problem. We shall make use of the reduced dual problem (2.4). Fix $\lambda > 0$. For any $\Lambda$, define the Moreau-Yosida regularization of $d_2$ at $\Lambda$ associated with $\lambda$ by

$$G_\lambda(\Lambda) = \min_{\Gamma \in \mathcal{F}} d_2(\Gamma) + \frac{1}{2\lambda} \|\Gamma - \Lambda\|^2_{\mathcal{F}}.$$
Using this definition and the definition of $d_2$ in (2.4), we obtain that

$$G_{\lambda}(\Lambda) = \min_{r \in \Gamma \subseteq \mu^2 I} \sup_{y} \left\{ \langle \Gamma, \mathcal{H}(y) \rangle - \frac{1}{2} \|A(y) - b\|^2 \right\} + \frac{1}{2\lambda} \|\Gamma - \Lambda\|^2_F,$$

$$= \sup_{y} \left\{ \min_{r \in \Gamma \subseteq \mu^2 I} \left\{ \langle \Gamma, \mathcal{H}(y) \rangle + \frac{1}{2\lambda} \|\Gamma - \Lambda\|^2_F \right\} - \frac{1}{2} \|A(y) - b\|^2 \right\},$$

where the second equality holds due to the compactness of the spectral norm ball [47, Corollary 37.3.2]. Furthermore, it is not hard to show that

$$\min_{r \in \Gamma \subseteq \mu^2 I} \left\{ \langle \mathcal{H}(y), \Gamma \rangle + \frac{1}{2\lambda} \|\Lambda - \Gamma\|^2_F \right\} = \langle \mathcal{H}(y), \Lambda \rangle - \frac{1}{2\lambda} \|\mathcal{H}(y)\|^2_F + \frac{1}{2\lambda} \|P_{\mu}(\Lambda - \lambda \mathcal{H}(y))\|^2_F,$$

where $P_{\mu}(W)$ is the unique optimal solution to the following convex optimization problem:

$$\min_{Z} \|Z\|_* + \frac{1}{2\mu} \|Z - W\|^2_F.$$

Thus, $G_{\lambda}(\Lambda) = \sup_{y} \Theta_{\lambda}(y; \Lambda),$ where

$$\Theta_{\lambda}(y; \Lambda) := \langle \mathcal{H}(y), \Lambda \rangle - \frac{1}{2\lambda} \|\mathcal{H}(y)\|^2_F + \frac{1}{2\lambda} \|P_{\mu}(\Lambda - \lambda \mathcal{H}(y))\|^2_F - \frac{1}{2} \|A(y) - b\|^2.$$

Recall from the Moreau–Yosida regularization theory that $P_{\mu}(\cdot)$ is globally Lipschitz continuous with modulus 1 and that $\|P_{\mu}(\cdot)\|^2_F$ is continuously differentiable with $\nabla(\|P_{\mu}(Y)\|^2_F) = 2P_{\mu}(Y)$. Hence, $\Theta_{\lambda}(\cdot; \Lambda)$ is a continuously differentiable concave function in $y$ with

$$\nabla_y \Theta_{\lambda}(y; \Lambda) = \mathcal{H}^*(\Lambda - \lambda \mathcal{H}(y)) - \mathcal{H}^*P_{\mu}(\Lambda - \lambda \mathcal{H}(y)) - A^*(A(y) - b).$$

In addition, we have that

$$\|\nabla_y \Theta_{\lambda}(y'; \Lambda) - \nabla_y \Theta_{\lambda}(y; \Lambda)\|_F$$

$$\leq (\lambda \|\mathcal{H}^*\mathcal{H}\| + \|A^*A\|) \|y' - y\|_F + \|\mathcal{H}^*P_{\mu}(\Lambda - \lambda \mathcal{H}(y')) - \mathcal{H}^*P_{\mu}(\Lambda - \lambda \mathcal{H}(y))\|_F$$

$$\leq (\lambda \|\mathcal{H}^*\mathcal{H}\| + \|A^*A\| + \lambda \|\mathcal{H}\|) \|y' - y\|_F,$$

which implies that $\nabla_y \Theta_{\lambda}(\cdot; \Lambda)$ is Lipschitz continuous with Lipschitz modulus

$$2\lambda \|\mathcal{H}^*\mathcal{H}\| + \|A^*A\| + \lambda \|\mathcal{H}\|.$$  \hspace{1cm} (3.1)

We are now ready to describe the PPA for solving the dual problem (2.3).

**Dual PPA.**

**Step 0.** Input $(y^0, \Lambda^0)$ and $\lambda_0 > 0$.

**Step 1.** (Find an approximate maximizer $y^{k+1} \approx \arg\max \Theta_{\lambda_k}(y; \Lambda^k).$)

Input $u^0 := y^k$. Let $s_k, \text{intol}_k > 0$, $1 > t, \sigma > 0$.

While $\|\nabla_y \Theta_{\lambda_k}(u^l; \Lambda^k)\| > \text{intol}_k$ do

(a) Let $s_l$ be the largest element of $\{s_k, t^2s_k, \ldots\}$ satisfying

$$\Theta_{\lambda_k}(u[s]; \Lambda^k) > \Theta_{\lambda_k}(u^l; \Lambda^k) + \sigma s \|\nabla_y \Theta_{\lambda_k}(u^l; \Lambda^k)\|^2_F,$$

where $u[s] = u^l + s\nabla_y \Theta_{\lambda_k}(u^l; \Lambda^k).$
(b) Set $u^{l+1} \leftarrow u[s_l]$, $l \leftarrow l + 1$.

End (while)

Set $y^{k+1} \leftarrow u^{k+1}$.

**Step 2.** Compute the SVD

$$\Lambda^k - \lambda_k H(y^{k+1}) = U \Sigma V^T.$$  

Set

$$\Lambda^{k+1} = U \min(\Sigma, \mu I)V^T.$$  

**Step 3.** If a termination criterion is not met, update $\lambda_k$. Go to Step 1.

### 3.3. Dual gradient projection methods.

In this section we assume that $A^*A$ is invertible. Recall that in this case, the dual of (2.1) is given by (2.5). Moreover, we have

$$\|\nabla d_2(\Lambda_1) - \nabla d_2(\Lambda_2)\|_F^2 = \|H((A^*A)^{-1}H^*(\Lambda_1 - \Lambda_2))\|_F^2 \leq r \|((A^*A)^{-1}H^*(\Lambda_1 - \Lambda_2))\|_F^2 \leq \left(\frac{r}{\sigma_{\min}(A^*A)}\right)^2 \|\Lambda_1 - \Lambda_2\|_F^2.$$  

This shows that the gradient of $d_2$ is Lipschitz continuous with Lipschitz constant

$$LD := \frac{r}{\sigma_{\min}(A^*A)}.$$  

Since the projection onto the feasible set of (2.5) is simple, we can apply the gradient projection (GP) methods to solve (2.5). One simple version is described as follows.

**Dual GP.**

**Step 0.** Input $\Lambda^0$ such that $\Lambda^0 \succeq \mu^2 I$ and $L > \frac{L_D}{2}$.  

**Step 1.** Compute the SVD

$$\Lambda^k - \frac{1}{L} \nabla d_2(\Lambda^k) = U \Sigma V^T,$$

where $U$ and $V$ have orthogonal columns, $\Sigma$ is diagonal. Set

$$\Lambda^{k+1} = U \min(\Sigma, \mu I)V^T.$$  

**Step 2.** If a termination criterion is not met, go to Step 1.

The iterate generated by the Dual GP satisfies

$$d_2(\Lambda^k) - v = \mathcal{O}\left(\frac{L}{k}\right);$$

see, e.g., [53, Theorem 1]. Hence, for faster convergence, a smaller $L$ is favored. Also, note that if $y^*$ and $\Lambda^*$ are solutions to (2.1) and (2.5) respectively, then we can see from (2.4) that

$$y^* = R(\Lambda^*) + \bar{b},$$

where $R(\Lambda) := (A^*A)^{-1}H^*(\Lambda)$ and $\bar{b} := (A^*A)^{-1}A^*b$. Hence, the sequence

$$y^k := R(\Lambda^k) + \bar{b} \quad (3.2)$$

converges to a solution to (2.1), which can be used to check for termination; see Section 4.1.

The Dual GP can be accelerated using Nesterov’s extrapolation techniques (see, e.g., [40–43,53]). This method has also been used in [45,51] for nuclear norm related problems. The method is described below.
Dual AGP.

**Step 0.** Input $\Lambda^0$ such that $\Lambda^0 \preceq \mu^2 I$ and $L \geq L_D$. Initialize $\Lambda^{-1} = \Lambda^0$ and $\theta^{-1} = \theta_0 = 1$. Go to Step 1.

**Step 1.** Set

$$\Psi^k = \Lambda^k + \left( \frac{\theta_k}{\theta_{k-1}} - \theta_k \right) (\Lambda^k - \Lambda^{k-1}).$$

Compute the SVD

$$\Psi^k - \frac{1}{L} \nabla d_2(\Psi^k) = U \Sigma V^T,$$

where $U$ and $V$ have orthogonal columns, $\Sigma$ is diagonal. Update

$$\Lambda^{k+1} = U \min\{\Sigma, \mu I\} V^T, \quad \theta_{k+1} = \sqrt{\theta_k^4 + 4\theta_k^2} \theta_k^2.$$

**Step 2.** If a termination criterion is not met, go to Step 1.

The sequence generated from the Dual AGP satisfies

$$d_2(\Lambda^k) - v = O\left(\frac{L}{k^2}\right);$$

see, e.g., [53, Theorem 1]. Hence, for faster convergence, a smaller $L$ is favored. To generate a suitable primal variable at each iteration, instead of setting $y^k = R(\Lambda^k) + \bar{b}$ as above, we can also initialize $y^k$ to be the zero matrix and update the sequence in Step 1 according to

$$y^{k+1} = (1 - \theta_k) y^k + \theta_k (R(\Psi^k) + \bar{b}).$$

For this choice of $y^k$, it can be shown, following the proof of [52, Corollary 2], that

$$0 \leq f(y^{k+1}) + d_2(\Lambda^{k+1}) = O(\theta_k^2).$$

Since $\theta_k \leq \frac{\sqrt{2}}{k+2}$, the duality gap falls below a given tolerance $tol > 0$ after at most $O\left(\sqrt{\frac{L}{\theta_0}}\right)$ iterations. In our implementation of Dual AGP in the next section, the latter choice of $y^k$ usually does not lead to faster convergence and thus we only report results using (3.2).

4. System identification. In this section, we consider the problem of identifying a linear dynamical system from observations of its inputs and outputs. Given a sequence of inputs $u_t \in \mathbb{R}^p$ and measured (noisy) outputs $\tilde{y}_t \in \mathbb{R}^m$, $t = 1, \ldots, N$, the goal is to find a discrete-time, linear time-invariant state space model,

$$x_{t+1} = Ax_t + Bu_t, \quad y_t = Cx_t + Du_t,$$

(4.1)

that satisfies $y_t \approx \tilde{y}_t$ and is low-order (i.e., corresponds to a low-dimensional state vector $x_t \in \mathbb{R}^r$). To determine the model, we need to find the $A, B, C, D$ matrices, the initial state $x_0$, and the model order $r$. As described in [31, Eq. (23)] (see also [32]), under reasonable assumptions, the minimal model order is equal to the rank of the matrix $H_{m,1,r+1,N+1-r}(y)U^\perp$, where $U^\perp \in \mathbb{R}^{(N+1-r) \times q}$ is a matrix whose columns
form an orthogonal basis of the null space of $H_{p;1,r+1,N+1-r}(u)$, and $u$ is the input to the system (see [31] for further details). Upon relaxing the rank function to nuclear norm, the tradeoff between the fitting error and the model order can be captured by the optimization problem,

$$\min_y \frac{1}{2} \|y - \tilde{y}\|_F^2 + \mu \|H_{m,1,r+1,N+1-r}(y)U^\perp\|_*, \quad (4.2)$$

where $\tilde{y} \in \mathbb{R}^{m \times (N+1)}$, $N \geq 1$, $r \geq 0$, $\mu > 0$, $H_{m,1,r+1,N+1-r}(y) = \begin{pmatrix} y_0 & y_1 & \cdots & y_{N-r} \\ y_1 & y_2 & \cdots & y_{N-r+1} \\ \vdots & \vdots & \ddots & \vdots \\ y_r & y_{r+1} & \cdots & y_N \end{pmatrix} \in \mathbb{R}^{m(r+1) \times (N+1-r)}$.

This problem corresponds to (2.1) with $\Upsilon = U^\perp$, $A(y) = \text{vec}(y)$ and $b = \text{vec}(\tilde{y})$. Thus $\sigma_{\text{max}}(A) = \sigma_{\text{max}}(\Upsilon) = 1$, $H^*(\Lambda) = H^*_{m,1,r+1,N+1-r}(A(U^\perp)^T)$.

Note that the solution set of (4.2) is nonempty since the function $y \mapsto \|y - \tilde{y}\|_F^2$ is coercive.

### 4.1. Computational results

In this section, we compare different algorithms for solving (4.2) on random and real data. Specifically, we consider Primal and Dual ADM, Dual PPA, Dual GP, and Dual AGP. Moreover, taking into account the fact that $A = I$, we also consider a variant of Primal ADM (referred to as Primal ADM2) by considering the following augmented Lagrangian function

$$\ell_\beta(Y, y, \Lambda) := \frac{1}{2} \|y - \tilde{y}\|_F^2 + \mu \|Y\|_* - \langle \Lambda, Y + \mathcal{H}(y) \rangle + \frac{\beta}{2} \|Y + \mathcal{H}(y)\|_F^2,$$

and replacing the minimization with respect to $y$ by

$$y^{k+1} := \arg \min_y \left\{ \ell_\beta(Y^{k+1}, y, \Lambda^k) + \frac{\beta}{2} \|y - y^k\|_Q^2 \right\},$$

where

$$Q_3 := \frac{1}{\sigma} I - H^*H > 0 \quad \text{and} \quad \sigma < \frac{1}{r},$$

so that $\sigma \|H^*H\| < 1$; as well as a variant of Dual ADM with $Q_1 = 0$ (referred to as Dual ADM2). The convergence of these two variants follows from Theorem B.1.

We initialize all algorithms except Dual PPA at the origin, where the latter algorithm is initialized at an approximate solution obtained from Dual AGP. We terminate the algorithms by checking relative duality gap with $\text{tol} = 1e-4$, i.e.,

$$\frac{\min\{f(y^k), f(\tilde{y}^k)\} + d(-H^*(P(\Lambda^k)))}{\max\{1,|d(-H^*(P(\Lambda^k)))|\}} < 1e-4 \quad \text{or} \quad \frac{f(y^k) + d_2(\Lambda^k)}{\max\{1,|d_2(\Lambda^k)|\}} < 1e-4,$$

We terminate Dual AGP by checking relative duality gap with $\text{tol} = 5e-3$, checked every 10 iterations. We also early terminate the algorithm if the change in Frobenius norm of successive iterates is small ($< 1e-8$ for each variable) or the maximum number of iteration hits 2000.
where \( \{ (y^k, \Lambda^k) \} \) are defined in each of Sections 3.1, 3.2 and 3.3, \( \mathcal{P}(\Lambda^k) \) is the projection of \( \Lambda^k \) onto the spectral norm ball with radius \( \mu \), and \( \hat{y}^k := y^k + \mathcal{H}^* (\mathcal{P}(\Lambda^k)) \) \(^3\). The termination criterion is checked every 10 iterations except for Dual PPA, where we do this every (outer) iteration, as the relative duality gap is required for updating \( \lambda_k \) and \( \text{intol}_k \). \(^4\) We also early terminate the algorithms if the change in Frobenius norm of successive iterates is small \((< 1e-8 \text{ for each variable}) \) or the maximum number of iteration hits 2000. We set \( \beta = \frac{\mu \sigma}{2 \sigma_{\text{max}}(y)} \) and \( \sigma = \frac{0.95}{r+1} \) for Primal ADM. We use the same \( \beta \) for Primal ADM2 and set \( \sigma = \frac{0.95}{r} \). Furthermore, we set \( \beta = \frac{\sigma_{\text{max}}(y)}{160r} \), \( \sigma_1 = 0.95 \) and \( \sigma_2 = \frac{0.95}{r} \) for Dual ADM and use the same \( \beta \) and \( \sigma_2 \) for Dual ADM2. We take \( L = \frac{1}{150} \) for Dual GP and \( L = L_D \) for Dual AGP. All codes are written in Matlab and run on a Sunfire X4440, with 4 Opteron 8384 quad-core CPUs and 32G of RAM, equipped with CentOS 5.8 and Matlab 7.12.

4.1.1. Random data. We generate randomly a matrix \( u = (u_0 \cdots u_N) \in \mathbb{R}^{p \times (N+1)} \) with standard Gaussian entries (i.e., zero mean and variance one), and let \( \bar{r} \) be the true order of the system. We then generate matrices \( A \in \mathbb{R}^{p \times r}, B \in \mathbb{R}^{p \times p}, C \in \mathbb{R}^{m \times r} \) and \( D \in \mathbb{R}^{m \times p} \) with i.i.d. standard Gaussian entries, and normalize them to have spectral norm 1. We also generate a vector \( x_0 \in \mathbb{R}^{r} \), again with standard Gaussian entries. The output \( \bar{y} = (\bar{y}_0 \cdots \bar{y}_N) \in \mathbb{R}^{m \times (N+1)} \) is then generated using a state-space model: for each \( t = 0, \ldots, N \),

\[
\begin{align*}
x_{t+1} &= Ax_t + Bu_t \\
\bar{y}_t &= Cx_t + Du_t.
\end{align*}
\]

To model the measurement noise, we add noise to the output \( \bar{y} \) to get \( \bar{\epsilon} = \bar{y} + \sigma \epsilon \), where \( \epsilon \) has Gaussian entries with mean 0 and variance 1, and \( \sigma > 0 \). Finally, \( U^{\perp} \) is a matrix whose columns form an orthonormal basis of the nullspace of \( H_{p,1,2r+2,N-2r}(u) \). Note that in theory, we require the \( r \) used in determining the size of the Hankel matrix to be larger than the true order of the system. However in practice, we often don’t know the true system order, and only have a guess or estimate for it. Therefore, when we set the size of the Hankel matrix in our problem, as a rule of thumb, we use roughly twice the estimated order; i.e., \( r = 2\bar{r} + 1 \).

In the tests below, we consider \( p = 5 \), \( m = 5, 10 \), \( r = 10, 20 \) and \( N + 1 = 2000, 4000 \). We pick \( \sigma = 5e - 2 \), which corresponds roughly to a 5% noise. The statistics of the test problems used are reported in Table 4.1\(^5\). We run our algorithms for \( \mu = 1e - 2, 1e - 1, 1, 10 \). We observed that Primal ADM2 usually outperforms Primal ADM, and the two Dual ADMs perform similarly. Thus, we do not report the performances of Primal ADM and Dual ADM.

Our computational results are reported in Table 4.2, where \( \text{iter} \) stands for the number of iterations, \( \text{cpu} \) is the CPU time taken and \( \text{obj} \) is the primal objective value at termination corresponding to each algorithm. The words “Primal” and “Dual” are abbreviated as “P.” and “D.” respectively. For Dual PPA, the CPU time and number

\(^3\)This is modeled after (3.2).

\(^4\)For Dual PPA, we set \( t = 0.3, \sigma = 1e - 4, s_0 = \frac{1e}{2}, \) with \( L \) given in (3.1). For each fixed \( k \) and any \( l \geq 1 \), we set \( s_l = 1.11s_{l-1} \) if \( s_{l-1} = s_k \), and \( s_l = s_{l-1} \) otherwise. The inner iterations are terminated when \( l \) hits 1000. The parameters \( \lambda_{k} \) are initialized at \( \lambda_0 = 1 \) and the update follows the rule that \( \lambda_{k+1} = 2\lambda_{k} \) if \( \text{gap}_k > 0.95\text{gap}_{k-1} \), where \( \text{gap}_k \) denotes the relative duality gap in the \( k \)th outer iteration. The \( \text{intol}_k \) decreases from 0.04 based on the value and the decrease of \( \text{gap}_k \), and is bounded below by \( 1e - 3 \).

\(^5\)Note that each of our algorithm requires an SVD of a matrix of size \( m(r+1) \times q \) in each iteration.
of iterations for Dual AGP used for initialization are in parenthesis, and the CPU time not in parenthesis refers to the total runtime. The word “max” denotes the maximum number of iterations. The fastest algorithm(s) in each instance is highlighted in bold. We see that the gradient projection algorithms usually work best when \( \mu \) is small, with Dual AGP usually more robust (i.e., solving all instances within 2000 iterations) than Dual GP. Furthermore, Dual PPA usually works best when \( \mu \) is large.

**Table 4.1**

Parameters for the randomly generated test problems.

<table>
<thead>
<tr>
<th>P</th>
<th>( N + 1 )</th>
<th>( \bar{r} )</th>
<th>m</th>
<th>q</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2000</td>
<td>10</td>
<td>5</td>
<td>1869</td>
</tr>
<tr>
<td>2</td>
<td>2000</td>
<td>10</td>
<td>10</td>
<td>1869</td>
</tr>
<tr>
<td>3</td>
<td>2000</td>
<td>20</td>
<td>5</td>
<td>1749</td>
</tr>
<tr>
<td>4</td>
<td>2000</td>
<td>20</td>
<td>10</td>
<td>1749</td>
</tr>
<tr>
<td>5</td>
<td>4000</td>
<td>10</td>
<td>5</td>
<td>3869</td>
</tr>
<tr>
<td>6</td>
<td>4000</td>
<td>10</td>
<td>10</td>
<td>3869</td>
</tr>
<tr>
<td>7</td>
<td>4000</td>
<td>20</td>
<td>5</td>
<td>3749</td>
</tr>
<tr>
<td>8</td>
<td>4000</td>
<td>20</td>
<td>10</td>
<td>3749</td>
</tr>
</tbody>
</table>

**4.1.2. Real data from DaISy database.** In this section, we consider eight benchmark problems from DaISy (Database for the Identification of Systems) [11]. A brief description of the data is given in Table 4.3. Each data set is given in form of a \( y \) (output) and \( u \) (input). We take the first 25% of the inputs and outputs for testing purposes. We set \( \bar{r} = 20 \) (hence, \( r = 2\bar{r} + 1 = 41 \)) and generate the matrix \( U^\perp \) so that the column space forms an orthonormal basis of the nullspace of \( H_{p,1,2r+2,N−2r}(u) \), where \( u = (u_0 \cdots u_N) \in \mathbb{R}^{p \times (N+1)} \). We apply the five algorithms from Table 4.2 to solve these problems for different values of \( \mu \), using the same parameters as in the previous section. The results are reported in Table 4.4, where \( \text{iter} \) and \( \text{cpu} \) are as in Table 4.2, while \( \text{nr} \) denotes the number of singular values of \( H(y^*) \) that are larger than \( 0.005 \sigma_{\text{max}}(H(y^*)) \), and \( \text{err} \) denotes the fitting error measured by \( \|y^* - \tilde{y}\|_F \); here, \( y^* \) is the approximate optimal solution of (4.2) obtained by the algorithms. We see that Dual AGP and Dual PPA work best. Furthermore, we see that \( \text{nr} \) decreases with \( \mu \), while \( \text{err} \) increases as \( \mu \) increases.

To further illustrate the performance of the first-order methods on real data, we consider the first problem in Table 4.3 and attempt to identify the system order using solutions obtained from Dual AGP. For comparison with the work [31], we use the same number of data points for identification and validation as in [31, Table 5.1]: i.e., 200 points for identification and 600 points for validation. We draw parameters \( \mu \) between \( 1e - 4 \) and 10, set \( r = \left\lfloor \frac{N+1}{p+1} \right\rfloor \) as in [31], solve the corresponding problem (4.2) by Dual AGP, then estimate the state-space matrices \( A, B, C, D \) and \( x_0 \) in (4.1) as done in [31, Section 5.2] and compute our identification errors (\( \text{err}_{\text{id}} \)) and validation errors (\( \text{err}_{\text{v}} \)) as in [31, Eq. (5.7)]:

\[
err_{\text{id}} = \sqrt{\frac{\sum_{t=0}^{199} \|\hat{y}_t - \tilde{y}_t\|^2}{\sum_{t=0}^{199} \|\hat{y}_t - \tilde{y}_t\|^2}}, \quad err_{\text{v}} = \sqrt{\frac{\sum_{t=0}^{599} \|\hat{y}_t - \tilde{y}_t\|^2}{\sum_{t=0}^{599} \|\hat{y}_t - \tilde{y}_v\|^2}},
\]

where

\[
\hat{y}_t = \frac{1}{200} \sum_{t=0}^{199} \hat{y}_t, \quad \tilde{y}_t = \frac{1}{600} \sum_{t=0}^{599} \tilde{y}_t.
\]
Approximate solution of (4.2) with $\mu$ $(y)$. The estimated system order is then set to be the rank of $\mu_y$. We identify the corresponding parameter $\mu_y$ is the output generated from (4.1) using the estimated $A$, $B$, $C$, $D$ and $x_0$. Table 4.3 shows some identification errors at which a "branching" occurs; see Table 4.6.

Table 4.2

<table>
<thead>
<tr>
<th>Description of test problems, taken from DaISy [11]</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>M.</strong></td>
</tr>
<tr>
<td>CD Player arm</td>
</tr>
<tr>
<td>Continuous Stirring Tank Reactor</td>
</tr>
<tr>
<td>Hair Dryer</td>
</tr>
<tr>
<td>Steam Heat Exchanger</td>
</tr>
<tr>
<td>Industrial Winding Process</td>
</tr>
<tr>
<td>Glass Furnace</td>
</tr>
<tr>
<td>Industrial Dryer</td>
</tr>
</tbody>
</table>

Table 4.3

<table>
<thead>
<tr>
<th>Description</th>
<th>N + 1</th>
<th>m</th>
<th>q</th>
</tr>
</thead>
<tbody>
<tr>
<td>CD Player arm</td>
<td>513</td>
<td>2</td>
<td>388</td>
</tr>
<tr>
<td>Continuous Stirring Tank Reactor</td>
<td>1876</td>
<td>2</td>
<td>1793</td>
</tr>
<tr>
<td>Hair Dryer</td>
<td>251</td>
<td>1</td>
<td>168</td>
</tr>
<tr>
<td>Steam Heat Exchanger</td>
<td>50/5.8/7.33e+0</td>
<td>14(9)/3.8(2.3)/4.67e+0</td>
<td></td>
</tr>
<tr>
<td>Industrial Winding Process</td>
<td>626</td>
<td>2</td>
<td>375</td>
</tr>
<tr>
<td>Glass Furnace</td>
<td>512</td>
<td>2</td>
<td>145</td>
</tr>
<tr>
<td>Industrial Dryer</td>
<td>217</td>
<td>3</td>
<td>50</td>
</tr>
</tbody>
</table>

and $\hat{y}_t$ is the output generated from (4.1) using the estimated $A$, $B$, $C$, $D$ and $x_0$.

We identify the corresponding parameter $\mu^*$ at which a "branching" occurs; see Figure 4.1, which plots identification and validation errors against the nuclear norm of $H(y)$. The estimated system order is then set to be the rank of $H(y^*)$, where $y^*$ is the approximate solution of (4.2) with $\mu = \mu^*$. Table 4.5 shows some identification errors.
Another commonly used heuristic for identifying system order is the subspace identification errors and validation errors for CD player arm data. From this and Figure 4.1, we conclude that the estimated system order in this case is 3, which agrees with the result obtained in [31, Table 5.2].
solution to (4.2) with lower accuracy than the interior point method used in [31], and the Matlab function n4sid has been updated since the publication of [31], it is worth repeating the comparison for the problems in Table 4.3. In the test below, we use the same number of data points for identification and validation as in [31, Table 5.1]. We report the system orders (ord), identification errors ($err_{id}$) and validation errors ($err_{v}$) obtained via the nuclear norm heuristics (nn), the n4sid with default order (n4sid$_d$), and the n4sid using the order obtained from the nuclear norm heuristics (n4sid$_n$) in Table 4.6. We observe that the orders and errors obtained from the nuclear norm heuristics are usually slightly different from those in [31, Table 5.2]. This can be attributed to the solution accuracy in solving (4.2), and the threshold for determining the rank of $H(y^*)$. On the other hand, we still observe that the nuclear norm heuristics usually yields lower identification and validation errors than the two versions of n4sid.\footnote{We observe that the errors for problems 2 and 4 obtained from n4sid are exceptionally large. In view of this, we also tried to vary the input order for n4sid from 1 to 12 for these two problems and observed the corresponding identification and validation errors. For problem 2, the smallest ($err_{id}, err_{v}$) = (0.19, 0.39), which occurs when order is 10; while for problem 4, the smallest ($err_{id}, err_{v}$) = (0.14, 0.22), which occurs when order is 8.}

\textbf{Comparison with interior point solver from [31].} Finally, we apply the interior point method in [31] to solve the first problem in Table 4.3 (i.e., with $N = 512$). We use the code (written in Python and calling cvxopt 1.1.3\footnote{The package cvxopt is installed from source using Python 2.6.5.}) available on the authors’
webpage, compiled with Python 2.6.5. While the interior point method provides a solution of higher accuracy (the relative duality gap is usually around $1 \times 10^{-6}$, rather than the $1 \times 10^{-4}$ as required in our first order methods), it is much slower. We observe that the number of iterations and CPU time for the interior point method are not sensitive to the change of $\mu$, and are usually around 9 iterations and 1000 seconds respectively. In particular, when $\mu \leq 1$, this method is significantly slower than Dual AGP, which takes between 1 and 40 seconds for this range of $\mu$. Thus, the first order methods appear more suitable for larger scale identification problems.

5. Stochastic system realization. Another fundamental problem in linear system theory is finding a minimal stochastic ARMA (autoregressive moving average) model for a vector random process, given noisy and/or partial estimates of process covariances. Covariance estimates are often obtained from sample averages of a sequence of noisy observations of the process, hence including both measurement noise and error due to the finite sample size. Here we focus on a form of this problem, described in [32, section II.B]; see also [34]. Consider a state-space model of an ARMA process $y_t \in \mathbb{R}^n$,

$$x_{t+1} = Ax_t + Be_t,$$
$$y_t = Cx_t + e_t,$$

where $x_t \in \mathbb{R}^r$, and $e_t$ is white noise with covariance matrix $Q$. The process covariances $h_i = \mathbb{E}(y ty_{t+i}^T)$ satisfy

$$h_0 = CPC^T + Q, \quad h_t = CA^{t-1}D, \quad t \geq 1,$$

where $D = APC^T + BQ$, and $P = \mathbb{E}(x_tx_t^T)$ satisfies the Lyapunov equation $P = APA^T + QBQ^T$. In a stochastic realization problem, we are given noisy estimates of $h_i$, denoted by $\tilde{h}_i$, $i = 1, \ldots, T-1$, and the goal is to find the minimal model order $r$ as well as the model parameters $A, B, C, Q$. It is known that the minimal order is equal to the rank of the block-Hankel matrix $H_{n,n,j,k}$ consisting of the exact process covariances [34] (as in the system ID problem, we need the Hankel matrix to be large enough, that is, $j$ and $k$ should be larger than the rank).

A general form of this problem, allowing for both noise and missing covariance information, can be stated as follows,

$$\min_y \frac{1}{2} \| w \circ (y - \bar{h}) \|^2_F + \mu \| H_{m,n,j,k}(y) \|_*,$$  \tag{5.1}
Thus, of iteration hits 2000. Fröbenius norm of successive iterates is small ($\sigma = 5$).

PPA, we use the same parameters as in the previous section. $\sigma$ and $Primal ADM$ and use the same $\beta$ with $\{\}

algorithm is initialized at an approximate solution obtained from Primal ADM3

for solving (5.1). Specifically, we consider Primal ADM, Dual ADM and Dual PPA.

Moreover, since the action of $(H^*H + A^*A)^{-1}$ and $(I + A^*A)^{-1}$ on vectors can be easily computed, we also consider a variant of Primal ADM (referred to as Primal ADM3) obtained from the following augmented Lagrangian function

$$
\frac{1}{2} \|w \circ (y - \tilde{h})\|_F^2 + \mu \|Y\|_* - \langle \Lambda, Y + H(y) \rangle + \frac{\beta}{2} \|Y + H(y)\|_F^2;
$$

as well as a variant of Dual ADM with $Q_1 = 0$ (referred to as Dual ADM3). The convergence of these two variants follows from [2, Proposition 4.2] and Theorem B.1, respectively. Furthermore, notice that the quadratic term in (5.1) is not strictly convex in $y$, the dual problem will then have additional linear constraints that makes gradient projection expensive computationally. Thus, we do not consider gradient projection algorithms for solving (5.1).

We initialize all algorithms except Dual PPA at the origin, where the latter algorithm is initialized at an approximate solution obtained from Primal ADM3, and terminate the algorithms by checking

$$
\max \left\{ \frac{f(y^k) + d(-w \circ H^*(P(A^k)))}{\max\{1, |d(-w \circ H^*(P(A^k)))|\}}, \frac{\|H^*(P(A^k)) - w \circ H^*(P(A^k))\|_F}{\max\{1, \|H^*(P(A^k))\|_F\}} \right\} < 1e^{-4},
$$

with $\{(y^k, A^k)\}$ defined as in Sections 3.1 and 3.2, and $P$ is the projection onto the spectral norm ball with radius $\mu$. For the ADMs, we set $\beta = \frac{\mu r}{2\sigma_{max}(h)}$ and $\sigma = \frac{0.95}{r+1}$ for Primal ADM and use the same $\beta$ for Primal ADM3. We set $\beta = \frac{\sigma_{max}(h)}{8\mu r}$, $\sigma_1 = 0.95$ and $\sigma_2 = \frac{0.95}{r}$ for Dual ADM and use the same $\beta$ and $\sigma_2$ for Dual ADM3. For Dual PPA, we use the same parameters as in the previous section.

Following [32, Section II(B)], we generate matrices $A \in \mathbb{R}^{r \times r}$, $B \in \mathbb{R}^{r \times n}$ and $C \in \mathbb{R}^{n \times r}$ with i.i.d. standard Gaussian entries (i.e., with mean 0 and variance 1). These matrices are then normalized to have spectral norm 1. We also randomly generate an initial state $x_0 \sim N(0, I)$ and noise vectors $e_t \sim N(0, I)$ for $t = 0, ..., T - 1$, again with i.i.d. standard Gaussian entries. We then generate an output $\tilde{y}_t, t = 0, ..., T - 1$, according to the state space model:

$$
x_{t+1} = Ax_t + Be_t, \\
\tilde{y}_t = Cx_t + e_t.
$$

\[\text{We terminate Primal ADM3 by checking relative duality gap and relative dual feasibility with } tol = 5e - 3, \text{ checked every 10 iterations. We also early terminate the algorithm if the change in Fröbenius norm of successive iterates is small (< 1e - 8 for each variable) or the maximum number of iteration hits 2000.}\]
To model measurement noise, we further add noise to $\tilde{y}$ and get $\tilde{y} = y + \sigma \epsilon$, where $\epsilon$ has i.i.d. Gaussian entries with mean 0 and variance 1. We then set, for each $i = 0, \ldots, k - 1$,

$$\tilde{h}_i = \frac{1}{T} \sum_{t=0}^{T-1-i} \tilde{y}_{t+i} \tilde{y}_t^T,$$

and $\tilde{h}_i$ is zero for $i \geq k$. Finally, set $w = (w_0 \cdots w_{j+k-2})$ such that $w_0 = \cdots = w_{k-1}$ equals the matrix of all ones, and zero otherwise. In the tests below, we consider $T = 1000$, $n = 20, 30$ and $k = 100, 200$. We use $r = 10$ and hence set $j = 21$. We further pick $\sigma = 5\epsilon^2$. The statistics of the test problems used are reported in Table 5.1; recall that $q = nk$ in this case. We run our algorithms for a range of values of $\mu$, namely $\mu = 10^{-2}, 10^{-1}, 1, 10$, in our simulations below to study the performance of the algorithms for different values of $\mu$. The computational results are reported in Table 5.2. We see that Primal ADM3 works best in all instances.

### Table 5.1
Parameters for the randomly generated test problems.

<table>
<thead>
<tr>
<th>$P$</th>
<th>$k$</th>
<th>$n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td>20</td>
</tr>
<tr>
<td>2</td>
<td>100</td>
<td>30</td>
</tr>
<tr>
<td>4</td>
<td>200</td>
<td>20</td>
</tr>
<tr>
<td>5</td>
<td>200</td>
<td>30</td>
</tr>
</tbody>
</table>

### Table 5.2
Computational results for problems from Table 5.1.

<table>
<thead>
<tr>
<th>$P$</th>
<th>$\mu$</th>
<th>P. ADM</th>
<th>P. ADM3</th>
<th>D. ADM3</th>
<th>D. PPA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>iter/cpu/obj</td>
<td>iter/cpu/obj</td>
<td>iter/cpu/obj</td>
<td>iter/cpu/obj</td>
</tr>
<tr>
<td>1</td>
<td>0.01</td>
<td></td>
<td>30/16.8/5.38e+0</td>
<td>30/16.8/5.38e+0</td>
<td>135/456.9/5.38e+0</td>
</tr>
<tr>
<td></td>
<td>0.10</td>
<td></td>
<td>110/61.6/2.73e+1</td>
<td>110/61.6/2.73e+1</td>
<td>53/174.3/2.73e+1</td>
</tr>
<tr>
<td></td>
<td>1.00</td>
<td></td>
<td>30/16.5/4.18e+1</td>
<td>160/78.7/4.18e+1</td>
<td>32/156.3/4.18e+1</td>
</tr>
<tr>
<td></td>
<td>10.00</td>
<td></td>
<td>7/4.37/4.38e+1</td>
<td>220/108.6/4.38e+1</td>
<td>17/4.37/4.38e+1</td>
</tr>
<tr>
<td>2</td>
<td>0.01</td>
<td></td>
<td>160/223.9/9.42e+0</td>
<td>160/223.9/9.42e+0</td>
<td>125/122.9/9.42e+0</td>
</tr>
<tr>
<td></td>
<td>0.10</td>
<td></td>
<td>60/125.8/4.99e+1</td>
<td>90/124.0/4.99e+1</td>
<td>136/100.9/4.99e+1</td>
</tr>
<tr>
<td></td>
<td>1.00</td>
<td></td>
<td>20/30.8/7.04e+1</td>
<td>140/193.9/7.04e+1</td>
<td>9/63.2/7.04e+1</td>
</tr>
<tr>
<td></td>
<td>10.00</td>
<td></td>
<td>7/10.6/7.16e+1</td>
<td>190/263/7.16e+1</td>
<td>1/7.69/7.16e+1</td>
</tr>
<tr>
<td>3</td>
<td>0.01</td>
<td></td>
<td>30/35.1/7.16e+0</td>
<td>30/35.1/7.16e+0</td>
<td>130/376.6/7.16e+0</td>
</tr>
<tr>
<td></td>
<td>0.10</td>
<td></td>
<td>80/110.8/4.99e+1</td>
<td>90/124.0/4.99e+1</td>
<td>136/100.9/4.99e+1</td>
</tr>
<tr>
<td></td>
<td>1.00</td>
<td></td>
<td>20/30.8/7.04e+1</td>
<td>140/193.9/7.04e+1</td>
<td>9/63.2/7.04e+1</td>
</tr>
<tr>
<td></td>
<td>10.00</td>
<td></td>
<td>7/10.6/7.16e+1</td>
<td>190/263/7.16e+1</td>
<td>1/7.69/7.16e+1</td>
</tr>
<tr>
<td>4</td>
<td>0.01</td>
<td></td>
<td>30/35.1/7.16e+0</td>
<td>30/35.1/7.16e+0</td>
<td>130/376.6/7.16e+0</td>
</tr>
<tr>
<td></td>
<td>0.10</td>
<td></td>
<td>80/110.8/4.99e+1</td>
<td>90/124.0/4.99e+1</td>
<td>136/100.9/4.99e+1</td>
</tr>
<tr>
<td></td>
<td>1.00</td>
<td></td>
<td>20/30.8/7.04e+1</td>
<td>140/193.9/7.04e+1</td>
<td>9/63.2/7.04e+1</td>
</tr>
<tr>
<td></td>
<td>10.00</td>
<td></td>
<td>7/10.6/7.16e+1</td>
<td>190/263/7.16e+1</td>
<td>1/7.69/7.16e+1</td>
</tr>
</tbody>
</table>

We also adapted Cadzow’s method [4], an approach based on alternating projections, that has been used in engineering applications for similar problems. We start by specifying an estimate $\tilde{r}$ of the order and $\tilde{\epsilon} > 0$ of the noise level. In each iteration,
we first project onto the set of matrices with a Hankel structure to obtain $H^k$, then
project onto the ball centered at $\tilde{h}$ with radius $\tilde{e}$ to obtain $Y^k$, and finally project onto
the (nonconvex) set of matrices with rank less than $\tilde{r}$ to obtain $R^k$. The algorithm is
terminated when
\[
\max\{\|H^k - Y^k\|_F, \|R^k - Y^k\|_F\} \leq \max\{1, \|H^k\|_F\} < 1e^{-3}.
\]
In the examples we tested, the performance and convergence of this algorithm is very
sensitive to the values of the two parameters, the noise level and the order estimate.
Indeed, it is known to be suboptimal as discussed in [10, Section V.A]. On the
other hand, the convex optimization approach presented here only depends on one
parameter ($\mu$), and the performance does not change significantly as $\mu$ varies.

Finally, we mention that another common approach to solving the system real-
ization problem is via subspace methods [34]. For a comparison between the nuclear
norm approach and the subspace method applied to this problem, we refer the readers
to [32, Section 2B], where problem (5.1) was solved via an interior-point solver.

6. Concluding remarks. As a systematic approach to capturing the tradeoff
between a model’s order and complexity with fitting (and validation) errors, a tradeoff
that commonly arises in diverse modeling applications, we studied the optimization
problem of minimizing the nuclear norm of matrices with linear structure, including
Hankel, Toeplitz, and moment structures. We then focused on first-order methods
for solving the resulting optimization problem. In our computational experiments,
the gradient projection method (accelerated by Nesterov’s extrapolation techniques)
and the dual proximal point algorithm usually outperform other first-order methods
in terms of CPU time for large and small regularization parameters, respectively,
for the system identification problem, while for the system realization problem, the
alternating direction method, as applied to a certain primal reformulation, usually
outperforms other first-order methods in terms of CPU time. In our tests, we also
observe that these methods outperform the interior point implementation proposed
in [31] for system identification problems.

Finally, we remark that most of the algorithms proposed in this work can be
easily adapted to solve (1.1) with a general linear structure $X = L(y)$, as long as
$L(y)$ can be computed efficiently and the norm of $L$ can be estimated. We chose to
focus our discussion on the Hankel structure to relate more closely to our motivating
applications discussed in the introduction, which mainly concern Hankel structure.

An interesting direction for future work on this problem is whether there are
conditions under which the nuclear norm heuristic can be theoretically guaranteed to
find the minimum rank solution. In order to make analogies with the existing low-rank
matrix recovery framework, we can consider the following problem: how many generic,
random linear measurements of a rank-$r$, $n \times n$ Hankel matrix suffice for correct
recovery of the Hankel matrix? If we ignore the Hankel structure, existing results
on recovery from random Gaussian measurements require $O(nr)$ measurements [6];
however, it is expected that the true number would be much lower due to the Hankel
structure.

Another future research direction involves applying our algorithms to a broader
range of applications identified in the introduction. Some of these problems have
further structure that the algorithms can exploit.

Appendix A. An alternative formulation. In this appendix, we describe
an alternative formulation for modeling the structured matrix rank minimization.
Instead of minimizing a least square fitting error, we constrain the difference $\mathbf{A}(y) - b$ in a set modeling uncertainty. The problem is then formulated as follows:

$$\min \|H(y)\|_* \text{ s.t. } \mathbf{A}(y) - b \in \Omega,$$

where $\Omega$ is the closed convex set modeling uncertainty. For instance, problem (1.2) can be modeled as (A.1) with $\Omega = [l_1, b_1] \times [l_2, b_2] \times \cdots \times [l_n, b_n]$.

As in Section 2, we can rewrite (A.1) as

$$v_1 := \min \|Y\|_* \text{ s.t. } Y + H(y) = 0, \quad z + \mathbf{A}(y) = b, \quad z \in \Omega.$$  

(A.2)

It is then not hard to show that the dual to (A.2) is equivalent to solving

$$-v_1 = \min s_{\Omega}(\gamma) - b^T \gamma \text{ s.t. } H^*(\Lambda) + A^*(\gamma) = 0, \Lambda^T \Lambda \preceq I,$$

(A.3)

where $s_{\Omega}(\gamma) := \max_{y \in \Omega} y^T \gamma$ is the support function of the set $\Omega$.

Unlike (2.3), the objective function of (A.3) is in general not differentiable, hence gradient projection methods are not easily applicable. However, the ADMs and the dual PPA can be suitably applied to solve (A.2) and (A.3). The efficiency in solving the subproblems depends on the specific form of $\Omega$.

**Appendix B. Convergence of the proximal alternating direction method.**

In this appendix, we provide a convergence proof of a proximal alternating direction method, which covers all versions of proximal ADMs used in this paper. Consider the convex optimization problem with the following separable structure

$$\min f(x) + g(y) \text{ s.t. } Ax + By = c,$$

(B.1)

where $f : \mathcal{X} \to (-\infty, +\infty]$ and $g : \mathcal{Y} \to (-\infty, +\infty]$ are closed proper convex functions, $A : \mathcal{X} \to \mathcal{Z}$ and $B : \mathcal{Y} \to \mathcal{Z}$ are linear operators, $\mathcal{X}$, $\mathcal{Y}$ and $\mathcal{Z}$ are real finite dimensional Euclidean spaces with inner product $\langle \cdot, \cdot \rangle$ and its induced norm $\| \cdot \|$. The proximal alternating direction method (proximal ADM) for solving (B.1) takes the following form:

**Proximal ADM**

**Step 0.** Input $(x^0, y^0, z^0) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$.

**Step 1.** Set

$$\begin{align*}
x^{k+1} &= \arg \min_{x \in \mathcal{X}} f(x) - \langle z^k, Ax \rangle + \frac{\lambda}{2} \|Ax + By^k - c\|^2 + \frac{1}{2} \|x - x^k\|^2_S, \\
y^{k+1} &= \arg \min_{y \in \mathcal{Y}} g(y) - \langle z^k, By \rangle + \frac{\lambda}{2} \|Ax^{k+1} + By - c\|^2 + \frac{1}{2} \|y - y^k\|^2_T, \\
z^{k+1} &= z^k - \lambda (Ax^{k+1} + By^{k+1} - c),
\end{align*}$$

(B.2)

where $\lambda > 0$, $S$ and $T$ are self-adjoint positive semidefinite operators on $\mathcal{X}$ and $\mathcal{Y}$ respectively.
Step 2. If a termination criterion is not met, go to Step 1.

When \( S = 0 \) and \( T = 0 \), the proximal ADM (B.2) reduces to the classical ADM introduced by Glowinski and Marroco [22] and Gabay and Mercier [21]. It was shown by Eckstein and Bertsekas [15] that the ADM, as a special case of Douglas-Rachford splitting [20], is actually an application of the proximal point algorithm on the dual problem by means of a specially-constructed splitting operator. Based on the same argument by further applying a change of variable to the operators, Eckstein [14] presented the first proximal ADM as in (B.2) with \( S = \mu_1 I \) and \( T = \mu_2 I \) for positive constants \( \mu_1 > 0 \) and \( \mu_2 > 0 \). Later, He et al. [25] further extended the idea of Eckstein [14] to monotone variational inequalities to allow \( \lambda, S, \) and \( T \) to be replaced by different parameters \( \lambda_k, S_k, \) and \( T_k \) in each iteration. The convergence results provided in [14] and [25] for the proximal ADM both need \( S \) and \( T \) to be positive definite, which limits the applications of the method. However, we notice that by slightly revising the proof provided by He et al. in [25], one may readily prove the following theorem. For clarity and completeness, we include a proof here.

For technical reasons, we consider the following constraint qualification:

**CQ** There exists \((x_0, y_0) \in \text{ri}(\text{dom } f \times \text{dom } g) \cap P\), where \( P \) is the constraint set in (B.1).

Under **CQ**, it follows from [47, Corollary 28.2.2] and [47, Corollary 28.3.1] that \((\bar{x}, \bar{y}) \in \mathcal{X} \times \mathcal{Y}\) is an optimal solution to problem (B.1) if and only if there exists a Lagrange multiplier \( \bar{\lambda} \in \mathcal{Z} \) such that

\[
A^* \bar{z} \in \partial f(\bar{x}), \quad B^* \bar{z} \in \partial g(\bar{y}), \quad A\bar{x} + B\bar{y} - c = 0,
\]

(B.3)

where \( \partial f \) and \( \partial g \) are the subdifferential mappings of \( f \) and \( g \) respectively. Moreover, any \( \bar{z} \in \mathcal{Z} \) satisfying (B.3) is an optimal solution to the dual problem of (B.1).

**Theorem B.1.** Assume that the solution set of (B.1) is nonempty and **CQ** holds. Let \( \{(x^k, y^k, z^k)\} \) be generated from the proximal ADM. Then \( \{(x^k, y^k)\} \) converges to an optimal solution to (B.1) and \( \{z^k\} \) converges to an optimal solution to the dual problem of (B.1) if one of the following conditions holds:

(a) \( f \) and \( g \) are strongly convex;
(b) \( f \) is strongly convex and \( B^* B + T \) is positive definite;
(c) \( g \) is strongly convex and \( A^* A + S \) is positive definite;
(d) \( S \) is positive definite and \( B \) is injective;
(e) \( T \) is positive definite and \( A \) is injective;
(f) \( S \) and \( T \) are positive definite.

**Proof.** We first show that the sequence generated by proximal ADM is bounded. Notice that the iteration scheme (B.2) of the proximal ADM can be rewritten as: for \( k = 0, 1, 2, \cdots \),

\[
\begin{align*}
0 & \in \partial f(x^{k+1}) - A^*[z^k - \lambda(Ax^{k+1} + By^{k+1} - c)] + S(x^{k+1} - x^k), \\
0 & \in \partial g(y^{k+1}) - B^*[z^k - \lambda(Ax^{k+1} + By^{k+1} - c)] + T(y^{k+1} - y^k), \\
z^{k+1} & = z^k - \lambda(Ax^{k+1} + By^{k+1} - c).
\end{align*}
\]

(B.4)

On the other hand, since the subdifferential mappings of the closed proper convex functions are maximal monotone [48, Theorem 12.17], there exist some constants \( \sigma_1, \sigma_2 \geq 0 \) such that

\[
\langle u - \bar{u}, x^{k+1} - \bar{x} \rangle \geq \sigma_1 \|x^{k+1} - \bar{x}\|^2, \quad \forall u \in \partial f(x^{k+1}), \bar{u} \in \partial f(\bar{x}),
\]

\[
\langle v - \bar{v}, y^{k+1} - \bar{y} \rangle \geq \sigma_2 \|y^{k+1} - \bar{y}\|^2, \quad \forall v \in \partial g(y^{k+1}), \bar{v} \in \partial g(\bar{y}),
\]

(B.5)
where $\sigma_1 > 0$ if $f$ is strongly convex, $\sigma_2 > 0$ if $g$ is strongly convex, and $(\bar{x}, \bar{y}, \bar{z})$ satisfies (B.3). Combining (B.5) with (B.3) and (B.4), we have

\[
(A^*[z^k - \lambda(Ax^{k+1} + By^k - c)] - S(x^{k+1} - x^k) - A^*\bar{z}, x^{k+1} - \bar{x}) \geq \sigma_1 \|x^{k+1} - \bar{x}\|^2,
\]
\[
(B^*[z^k - \lambda(Ax^{k+1} + By^{k+1} - c)] - T(y^{k+1} - y^k) - B^*\bar{y}, y^{k+1} - \bar{y}) \geq \sigma_2 \|y^{k+1} - \bar{y}\|^2.
\]

Let \(w^k_c := w^k - \bar{w}\) for notational simplicity, where \(w\) represents \(x, y, z\) respectively. Then the two inequalities above can be rewritten as follows:

\[
\langle \bar{z}^k, Ax^{k+1} + By^k \rangle - \langle \lambda(Ax^{k+1} + By^k), Ax^{k+1} \rangle \geq \sigma_1 \|Ax^{k+1}\|^2,
\]
\[
\langle \bar{z}^k, Ax^{k+1} + By^k \rangle - \langle S(x^{k+1} - x^k), x^{k+1} \rangle \geq \sigma_1 \|Ax^{k+1}\|^2.
\]

Adding up these two inequalities we obtain that

\[
\langle \bar{z}^k, Ax^{k+1} + By^k \rangle - \lambda(Ax^{k+1} + By^k, Ax^{k+1}) - \langle S(x^{k+1} - x^k), x^{k+1} \rangle - \langle T(y^{k+1} - y^k), y^{k+1} \rangle \geq \sigma_1 \|Ax^{k+1}\|^2 + \sigma_2 \|y^{k+1}\|^2.
\]

Using the relation \(z^{k+1} = z^k - \lambda(Ax^{k+1} + By^k)\) and the elementary relations

\[\langle u, v \rangle = \frac{1}{2}(\|u\|^2 + \|v\|^2 - \|u - v\|^2) = \frac{1}{2}(\|u + v\|^2 - \|u\|^2 - \|v\|^2),\]

we obtain further that

\[
\|x^{k+1}\|^2 + \|y^{k+1}\|^2 + \lambda\|By^{k+1}\|^2 + \frac{1}{\lambda}\|z^{k+1}\|^2 \leq \|x^k\|^2 + \|y^k\|^2 + \lambda\|By^k\|^2 + \frac{1}{\lambda}\|z^k\|^2
\]

\[= (2\sigma_1\|x^{k+1}\|^2 + 2\sigma_2\|y^{k+1}\|^2 + \|x^{k+1} - x^k\|^2 + \|y^{k+1} - y^k\|^2 + \lambda\|Ax^{k+1} + By^k\|^2).
\]

(B.6)

From (B.6), we see immediately that the sequence \(\{\|x^k\|^2, \|y^k\|^2 + \lambda\|By^k\|^2 + \frac{1}{\lambda}\|z^k\|^2\}\) is monotonically nonincreasing (and thus bounded), and

\[
\lim_{k \to \infty} 2\sigma_1\|x^{k+1}\|^2 + 2\sigma_2\|y^{k+1}\|^2 + \|x^{k+1} - x^k\|^2 + \|y^{k+1} - y^k\|^2 + \lambda\|Ax^{k+1} + By^k\|^2 = 0.
\]

(B.7)

As an immediate consequence of these two facts, we see that the sequences

\[
\{\|x^{k+1}\|^2\}, \{\|y^{k+1}\|^2\}, \{\|By^{k+1}\|^2\}, \{\|z^{k+1}\|^2\},
\]

\[
\{\sigma_1\|x^{k+1}\|^2\}, \{\sigma_2\|y^{k+1}\|^2\}, \{\|Ax^{k+1} + By^k\|^2\}
\]

are bounded. Thus, the sequence \(\{\|By^{k+1}\|^2 + \|y^{k+1}\|^2\}\) is bounded. Moreover, since

\[
\|Ax^{k+1}\|^2 \leq \|Ax^{k+1} + By^k\|^2 + \|By^k\|^2,
\]

it follows that the sequence \(\{\|Ax^{k+1}\|^2 + \|z^{k+1}\|^2\}\) is also bounded. From the boundedness of the above sequences, it is now routine to deduce that the sequence \(\{x^k, y^k, z^k\}\) is bounded under the assumption of the theorem.

Since the sequence \(\{x^k, y^k, z^k\}\) is bounded, there exists a subsequence \(\{x^{k_i}, y^{k_i}, z^{k_i}\}\) that converges to a limit point, say \((x^\infty, y^\infty, z^\infty)\). We next show that \((x^\infty, y^\infty)\) is an optimal solution to problem (B.1) and \(z^\infty\) is a corresponding Lagrange multiplier.

To this end, we first note from (B.7) that

\[
\lim_{k \to \infty} \sigma_1\|x^{k+1}\| = 0, \lim_{k \to \infty} \sigma_2\|y^{k+1}\| = 0, \lim_{k \to \infty} \|Ax^{k+1} + By^k\| = 0,
\]

\[
\lim_{k \to \infty} \|x^{k+1} - x^k\| = 0, \lim_{k \to \infty} \|y^{k+1} - y^k\| = 0.
\]

(B.9)
Thus, if either $S$ or $T$ is positive definite, we have
\[
\lim_{k \to \infty} \|x^{k+1} - x^k\| = 0 \quad \text{or} \quad \lim_{k \to \infty} \|y^{k+1} - y^k\| = 0.
\] (B.10)

Similarly, if either $f$ or $g$ is strongly convex, then (B.10) also holds. Thus, (B.10) holds under the assumption of the theorem. Since
\[
\|Ax_c^{k+1} + By_c^{k+1}\| \leq \|Ax_c^{k+1} + By_c^k\| + \|B(y^{k+1} - y^k)\|
\]
and
\[
\|Ax_c^{k+1} + By_c^{k+1}\| \leq \|Ax_c^{k+2} + By_c^{k+1}\| + \|A(x^{k+2} - x^{k+1})\|
\]
we obtain from (B.9), (B.10) and invoking the closedness of the graphs of $\partial f$ and $\partial g$ [3, Page 80], we obtain that
\[
\lim_{k \to \infty} \|z^{k+1} - z^k\| = \lim_{k \to \infty} \lambda \|Ax_c^{k+1} + By_c^{k+1}\| = 0.
\] (B.11)

Taking limits on both sides of (B.4) along the subsequence $\{(x^{k_i}, y^{k_i}, z^{k_i})\}$, using (B.9), (B.11) and invoking the closedness of the graphs of $\partial f$ and $\partial g$ [3, Page 80], we obtain that
\[
A^*z^\infty \in \partial f(x^\infty), \quad B^*z^\infty \in \partial g(y^\infty), \quad Ax^\infty + By^\infty - c = 0,
\]
i.e., $(x^\infty, y^\infty, z^\infty)$ satisfies (B.3). From the discussion prior to the theorem, we conclude that $(x^\infty, y^\infty)$ is an optimal solution to problem (B.1) and $z^\infty$ is a corresponding Lagrange multiplier.

Finally, we show that $(x^\infty, y^\infty, z^\infty)$ is actually the unique limit of $\{(x^{k_i}, y^{k_i}, z^{k_i})\}$ and hence complete the proof. To this end, recall that $(x^\infty, y^\infty, z^\infty)$ satisfies (B.3). Hence, we could replace $(\bar{x}, \bar{y}, \bar{z})$ with $(x^\infty, y^\infty, z^\infty)$ in the above arguments, starting from (B.5). As a consequence, the subsequence $\{\|x^{k_i}\|^2_S + \|y^{k_i}\|^2_T + \lambda \|By^{k_i}\|^2 + \frac{1}{\lambda} \|z^{k_i}\|^2\}$ now converges to 0 as $i \to \infty$. Since this sequence is also non-increasing, we conclude that the whole sequence converges to zero, i.e.,
\[
\lim_{k \to \infty} \|x^{k_i}\|^2_S + \|y^{k_i}\|^2_T + \lambda \|By^{k_i}\|^2 + \frac{1}{\lambda} \|z^{k_i}\|^2 = 0.
\] (B.12)

From this, we see immediately that $\lim_{k \to \infty} z^k = z^\infty$. Moreover, if $g$ is strongly convex or $B^*B + T$ is positive definite, then we see from (B.12) and (B.9) that
\[
\lim_{k \to \infty} \sigma_2 \|y^{k+1}\| = 0 \quad \text{and} \quad \lim_{k \to \infty} \|y^{k+1}\|^2_T + \|By^{k+1}\|^2 = 0,
\]
and hence $\lim_{k \to \infty} y^k = y^\infty$. On the other hand, if $f$ is strongly convex or $A^*A + S$ is positive definite, then we see from (B.12), (B.8) and (B.9) that
\[
\lim_{k \to \infty} \sigma_1 \|x^{k+1}\| = 0 \quad \text{and} \quad \lim_{k \to \infty} \|x^{k+1}\|^2_S + \|Ax^{k+1}\|^2 = 0,
\]
and hence $\lim_{k \to \infty} x^k = x^\infty$. Therefore, we have shown that the whole sequence $\{(x^{k_i}, y^{k_i}, z^{k_i})\}$ converges to $(x^\infty, y^\infty, z^\infty)$ under the assumption of the theorem. \[\]

Remark B.1. Without the two terms $\sigma_1 \|x^{k+1}\|^2$ and $\sigma_2 \|y^{k+1}\|^2$, the inequality (B.6) is actually a special case of the one used in Theorem 1 in [25].

Remark B.2. The conditions (a)–(f) for the convergence of $\{(x^{k_i}, y^{k_i}, z^{k_i})\}$ in Theorem B.1 can be replaced by assuming that the following two slightly stronger conditions hold:

(i) $f$ is strongly convex, or $A$ is injective, or $S$ is positive definite;
(ii) $g$ is strongly convex, or $B$ is injective, or $T$ is positive definite; except for the case that both $A$ and $B$ are injective.

**Remark B.3.** Theorem B.1 provides general conditions for the convergence of the proximal ADM. It includes some recent results in the literature as special cases. For example, it has been considered by Attouch and Soueycatt [1] for the case $S = \frac{1}{\lambda} I$ and $T = \frac{1}{\mu} I$. Moreover, Zhang et al. considered the case when $B = I$ and $S$ is chosen to be positive definite in [57], and established a slightly weaker convergence result, that is, all the cluster points of $\{(x^k, y^k)\}$ and $\{z^k\}$ are optimal solutions to the primal and dual problems, respectively. Furthermore, Yang and Zhang [56] applied the proximal ADM to solve $l_1$-norm minimization problems in compressive sensing, allowing the step length to be chosen under some conditions rather than only 1. It is notable that when the step length is chosen to be 1, the iteration scheme in [56] simply reduces to (B.2) with $A = I$ and a symmetric and positive definite matrix $T$. The proof of the convergence in [56] is very similar to the one provided in this appendix but particularly for the $l_1$-norm minimization problems.

**Acknowledgement.** The first and second authors would like to thank Zhang Liu and Lieven Vandenberghe for many useful discussions. The third author would like to thank his PhD student Mr. Weimin Miao at the National University of Singapore for discussions on the convergence of alternating direction methods. We would also like to thank the anonymous referees for their many suggestions that help improve the manuscript.

**REFERENCES**


